

Documentație: Arhitectura web cu Reverse Proxy, API Gateway și Load Balancer

Reverse Proxy

Un reverse proxy este un server intermediar care se află între clienți (utilizatorii finali) și unul sau mai multe servere backend (unde rulează aplicația).

- **Funcție:** Primește cererile de la clienți și le redirecționează către serverul backend corespunzător, apoi returnează răspunsul serverului către client.
 - **Beneficii:**
 - **Securitate sporită:** Ascunde identitatea serverelor backend, protejându-le de atacuri.
 - **Distribuția sarcinii:** Poate redirecționa traficul către servere diferite, deși un load balancer este mai specializat în acest sens.
 - **Terminare SSL:** Poate prelua traficul criptat SSL, reducând sarcina pe serverele de aplicație.
 - **Caching:** Poate stoca în cache conținutul static, îmbunătățind viteza de încărcare.
-

API Gateway

Un API gateway servește ca un punct de intrare unic pentru un set de servicii, în special în arhitecturile de microservicii.

- **Funcție:** Unifică multiplele servicii într-o singură interfață pentru clienți, simplificând comunicarea cu o arhitectură complexă.
 - **Beneficii:**
 - **Rutare:** Direcționează cererile de la client către serviciul corect.
 - **Autentificare și autorizare:** Gestionează securitatea, verificând permisiunile înainte de a redirecționa cererea.
 - **Monitorizare și limitare de trafic (rate limiting):** Monitorizează și controlează numărul de cereri, prevenind suprasolicitarea sistemului.
-

Load Balancer

Un load balancer (echilibrator de sarcină) distribuie cererile de rețea pe un grup de servere backend pentru a asigura disponibilitatea și fiabilitatea.

- **Funcție:** Trimite cererile doar către serverele care funcționează corect (verificând "starea de sănătate" a acestora) și poate adăuga sau elimina dinamic servere în funcție de volumul de trafic.
- **Algoritmi de distribuție:** Utilizează diverse strategii pentru a distribui traficul:

- **Round robin:** Trimite cererile în mod secvențial către fiecare server.
 - **Least connections:** Direcționează cererile către serverul cu cele mai puține conexiuni active.
 - **IP hash:** Direcționează întotdeauna cererile de la aceeași adresă IP către același server.
-

Cum colaborează aceste componente

Deși sunt independente, aceste trei componente lucrează adesea împreună pentru a crea o arhitectură web robustă.

- **Exemplu AWS:** O configurație comună pe Amazon Web Services (AWS) implică utilizarea unui **Amazon API Gateway** ca punct de intrare unic. Acesta rotează traficul către diverse microservicii, fiecare dintre ele fiind protejat și echilibrat de un **Elastic Load Balancer**.
- **Exemplu NGINX:** NGINX este un software care poate acționa atât ca **reverse proxy**, cât și ca **load balancer**. Poate fi configurat pentru a sta în fața serverelor de aplicație, distribuind cererile și oferind funcționalități suplimentare de caching și securitate.